

Digital twins of nonlinear dynamical systems

Ling-Wei Kong

Arizona State University <https://orcid.org/0000-0002-8921-1642>

Yang Weng

Arizona State University

Bryan Glaz

Army Research Laboratory

Mulugeta Haile

Army Research Laboratory

Ying-Cheng Lai (✉ Ying-Cheng.Lai@asu.edu)

Ying-Cheng.Lai@asu.edu <https://orcid.org/0000-0002-0723-733X>

Article

Keywords:

Posted Date: July 27th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1871210/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

VISIT...

LANZAROTE
Caliente.COM

Digital twins of nonlinear dynamical systems

Ling-Wei Kong,¹ Yang Weng,¹ Bryan Glaz,² Mulugeta Haile,² and Ying-Cheng Lai^{1,3,*}

¹*School of Electrical, Computer and Energy Engineering,
Arizona State University, Tempe, Arizona 85287, USA*

²*Vehicle Technology Directorate, CCDC Army Research Laboratory,
2800 Powder Mill Road, Adelphi, MD 20783-1138, USA*

³*Department of Physics, Arizona State University, Tempe, Arizona 85287, USA*

(Dated: July 18, 2022)

We articulate the design imperatives for machine-learning based digital twins for nonlinear dynamical systems subject to external driving, which can be used to monitor the “health” of the target system and anticipate its future collapse. We demonstrate that, with single or parallel reservoir computing configurations, the digital twins are capable of challenging forecasting and monitoring tasks. Employing prototypical systems from climate, optics and ecology, we show that the digital twins can extrapolate the dynamics of the target system to certain parameter regimes never experienced before, make continual forecasting/monitoring with sparse real-time updates under nonstationary external driving, infer hidden variables and accurately predict their dynamical evolution, adapt to different forms of external driving, and extrapolate the global bifurcation behaviors to systems of some different sizes. These features make our digital twins appealing in significant applications such as monitoring the health of critical systems and forecasting their potential collapse induced by environmental changes.

INTRODUCTION

The concept of digital twins originated from aerospace engineering for aircraft structural life prediction [1]. In general, a digital twin can be used for predicting dynamical systems and generating solutions of emergent behaviors that can potentially be catastrophic [2]. Digital twins have attracted a great deal of attention from a wide range of fields [3] including medicine and health care [4, 5]. For example, the idea of developing medical digital twins in viral infection through a combination of mechanistic knowledge, observational data, medical histories, and artificial intelligence has been proposed recently [6], which can potentially lead to a powerful addition to the existing tools to combat future pandemics. In a more dramatic development, the European Union plans to fund the development of digital twins of Earth for its green transition [7, 8].

The physical world is nonlinear. Many engineering systems, such as complex infrastructural systems, are governed by nonlinear dynamical rules, too. In nonlinear dynamics, various bifurcations leading to chaos and system collapse can take place [9]. For example, in ecology, environmental deterioration caused by global warming can lead to slow parameter drift towards chaos and species extinction [10, 11]. In an electrical power system, voltage collapse can occur after a parameter shift that lands the system in transient chaos [12]. The various climate systems in different geographic regions of the world are also nonlinear and the emergent catastrophic behaviors as the result of increasing human activities are of grave concern. In all these cases, it is of interest to develop

a digital twin of the system of interest to monitor its “health” in real time as well as for predictive problem solving in the sense that, if the digital twin indicates a possible system collapse in the future, proper control strategies should and can be devised and executed in time to prevent the collapse.

What does it take to create a digital twin for a nonlinear dynamical system? For natural and engineering systems, there are two general approaches: one is based on mechanistic knowledge and another is based on observational data. In principle, if the detailed physics of the system is well understood, it should be possible to construct a digital twin through mathematical modeling. However, there are two difficulties associated with this modeling approach. First, a real-world system can be high-dimensional and complex, preventing the rules governing its dynamical evolution from being known at a sufficiently detailed level. Second, the hallmark of chaos is sensitive dependence on initial conditions. Because no mathematical model of the underlying physical system can be perfect, the small deviations and high dimensionality of the system coupled with environmental disturbances can cause the model predictions of the future state of the system to be inaccurate and completely irrelevant [13, 14]. These difficulties motivate the proposition that the model-based approach may not be more advantageous than the purely data-based approach and a viable method to develop a digital twin is through data. While in certain cases, approximate system equations can be found from data through sparse optimization [15–17], the same difficulties with the modeling approach arise. These considerations have led us to exploit machine learning to create digital twins for nonlinear dynamical systems.

Given a nonlinear dynamical system, its digital twin is also a dynamical system, rendering appropriate exploita-

* Ying-Cheng.Lai@asu.edu

tion of recurrent neural networks that can be designed to generate self-dynamical evolution with memory. In this regard, reservoir computers (RC) [18–20] that have been extensively studied in recent years [21–42] provide a starting point, which can be trained from observational data to generate closed-loop dynamical evolution that follows the evolution of the target system for a finite amount of time. Another advantage of RC is that no backpropagation is needed for optimizing the parameters - only a linear regression is required in the training so it is computationally efficient. A common situation is that the target system is subject to external driving, such as a driven laser, a regional climate system, or an ecosystem under external environmental disturbances. Accordingly, the digital twin must accommodate a mechanism to control or steer the dynamics of the RC neural network to account for the external driving. Introducing a control mechanism distinguishes our work from existing ones in the literature of RC as applied to nonlinear dynamical systems. Of particular interest is whether the collapse of the target chaotic system can be anticipated from the digital twin. The purpose of this paper is to demonstrate that the digital twin so created can accurately produce the bifurcation diagram of the target system and faithfully mimic its dynamical evolution from a statistical point of view. The digital twin can then be used to monitor the present and future “health” of the system. More importantly, with proper training from observational data the twin can reliably anticipate system collapses, providing early warnings of potentially catastrophic failures of the system.

More specifically, using three prototypical systems from optics, ecology, and climate, respectively, we demonstrate that the RC based digital twins developed in this paper solve the following challenging problems: (1) extrapolation of the dynamical evolution of the target system into certain “uncharted territories” in the parameter space, (2) long-term continual forecasting of nonlinear dynamical systems subject to nonstationary external driving with sparse state updates, (3) inference of hidden variables in the system and accurate prediction of their dynamical evolution into the future, (4) adaptation to external driving of different waveforms, and (5) extrapolation of the global bifurcation behaviors of network systems to some different sizes. These features make our digital twins appealing in applications.

RESULTS

For clarity, we present results on the digital twin for a prototypical nonlinear dynamical systems with adjustable phase-space dimension: the Lorenz-96 climate network model [43]. In Supplementary Information, we present two additional examples: a chaotic laser (Supplementary Note 1 - SN 1) and a driven ecological system (SN 2), together with a number of pertinent issues.

A low-dimensional Lorenz-96 climate network and its digital twin. The Lorenz-96 system [43] is an idealized atmospheric climate model. Mathematically, the toy climate system is described by m coupled first-order nonlinear differential equations subject to external periodic driving $f(t)$:

$$\frac{dx_k}{dt} = x_{k-1}(x_{k+1} - x_{k-2}) + f(t), \quad (1)$$

where $k = 1, \dots, m$, is the spatial index. Under the periodic boundary condition, the m nodes constitute a ring network, where each node is coupled to three neighboring nodes. To be concrete, we set $m = 6$ (more complex high-dimensional cases are treated below). The driving force is sinusoidal with a bias F : $f(t) = A \sin(\omega t) + F$. We fix $\omega = 2$ and $F = 2$, and use the forcing amplitude A as the bifurcation parameter. For relatively large values of A , the system exhibits chaotic behaviors, as exemplified in Fig. 1(A1) for $A = 2.2$. Quasiperiodic dynamics arise for smaller values of A , as exemplified in Fig. 1(A2). As A decreases from a large value, a critical transition from chaos to quasiperiodicity occurs at $A_c \approx 1.9$. We train the digital twin with time series from four values of A , all in the chaotic regime: $A = 2.2, 2.6, 3.0$, and 3.4 . The size of the random reservoir network is $D_r = 1,200$. For each value of A in the training set, the training and validation lengths are $t = 2,500$ and $t = 12$, respectively, where the latter corresponds to approximately five Lyapunov times. The warming-up length is $t = 20$ and the time step of the reservoir dynamical evolution is $\Delta t = 0.025$. The hyperparameter values (See Sec. for their meanings) are optimized to be $d = 843$, $\lambda = 0.48$, $k_{\text{in}} = 0.29$, $k_c = 0.113$, $\alpha = 0.41$, and $\beta = 1 \times 10^{-10}$. Our computations reveal that, for the deterministic version of the Lorenz-96 model, it is difficult to reduce the validation error below a small threshold. However, adding an appropriate amount of noise into the training time series [18] can lead to smaller validation errors. We add an additive Gaussian noise with standard deviation σ_{noise} to each input data channel to the reservoir network [including the driving channel $f(t)$]. The noise amplitude σ_{noise} is treated as an additional hyperparameter to be optimized. For the toy climate system, we test several noise levels and find the optimal noise level giving the best validating performance: $\sigma_{\text{noise}} \approx 10^{-3}$.

Figures 1(B1) and 1(B2) show the dynamical behaviors generated by the digital twin for the same values of A as in Figs. 1(A1) and 1(A2), respectively. It can be seen that not only does the digital twin produce the correct dynamical behavior in the same chaotic regime where the training is carried out, it can also extrapolate beyond the training parameter regime to correctly predict the unseen system dynamics there (quasiperiodicity in this case). To provide support in a broader parameter range, we calculate true bifurcation diagram, as shown in Fig. 1(C), where the four vertical dashed lines indicate the four values of the training parameter. The bifurcation generated by the digital twin is shown in Fig. 1(D),

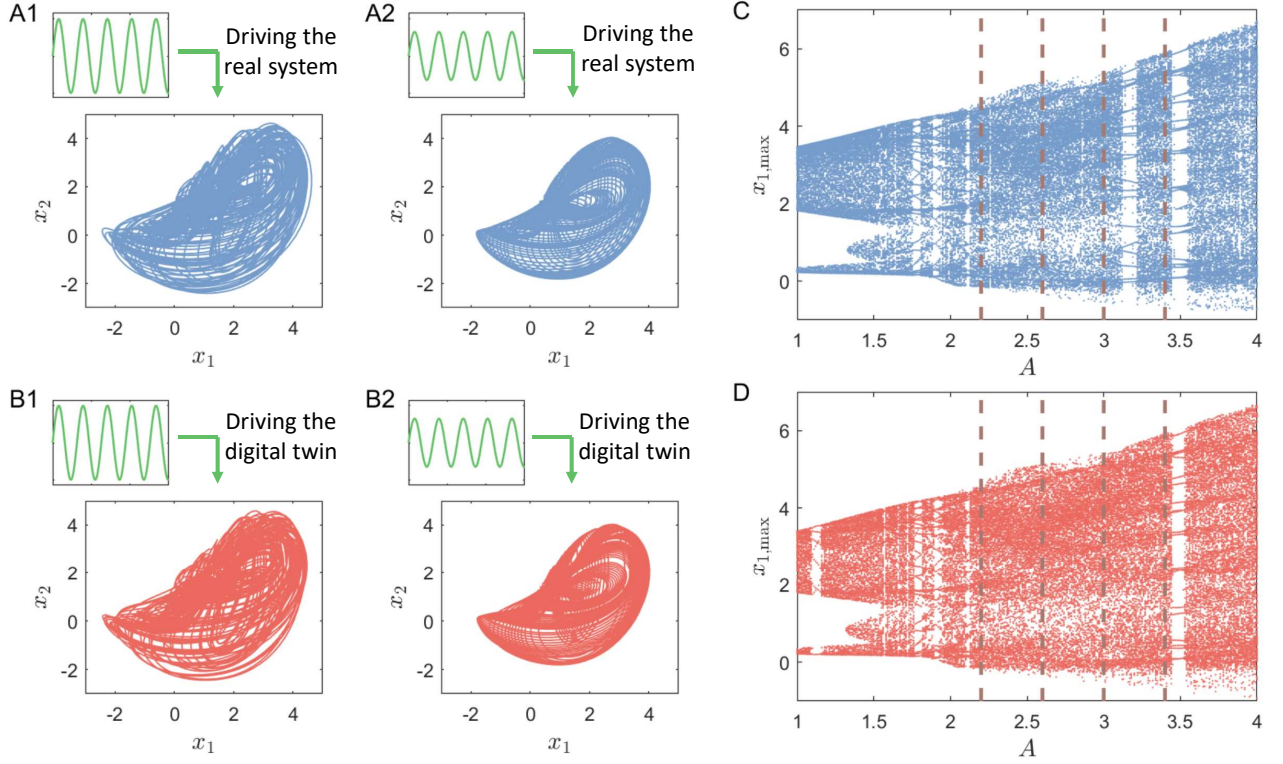


FIG. 1. Digital twin of the Lorenz-96 climate system. The toy climate system is described by six coupled first-order nonlinear differential equations (phase-space dimension $m = 6$), which is driven by a sinusoidal signal $f(t) = A \sin(\omega t) + F$. (A1,A2) Ground truth: chaotic and quasiperiodic dynamics in the system for $A = 2.2$ and $A = 1.6$, respectively, for $\omega = 2$ and $F = 2$. The sinusoidal driving signals $f(t)$ are schematically illustrated. (B1, B2) The corresponding dynamics of the digital twin under the same driving signal $f(t)$. Training of the digital twin is conducted using time series from the chaotic regime. The result in (B2) indicates that the digital twin is able to extrapolate outside the chaotic regime to generate the unseen quasiperiodic behavior. (C, D) True and digital-twin generated bifurcation diagrams of the toy climate system, where the four vertical red dashed lines indicate the values of driving amplitudes A , from which the training time series data are obtained. The remarkable agreement between the two bifurcation diagrams attests to the strong ability of the digital twin to reproduce the distinct dynamical behaviors of the target climate system in different parameter regimes, even with training data only in the chaotic regime.

which agrees remarkably well with the true diagram even at a detailed level.

Previously, it was suggested that RC can have a certain degree of extrapolability [34–39]. Figure 1 represents an example where the target system’s response is extrapolated to external sinusoidal driving with unseen amplitudes. In general, extrapolation is a difficult problem. Some limitations of the extrapolability with respect to the external driving signal is discussed in SN 1, where the digital twin can predict the crisis point but cannot extrapolate the asymptotic behavior after the crisis.

In the following, we systematically study the applicability of the digital twin in solving forecasting problems in more complicated situations than the basic settings demonstrated in Fig. 1. The issues to be addressed are high dimensionality, the effect of the waveform of the driving on forecasting, and the generalizability across Lorenz-96 networks of different sizes. Results of continual forecasting and inferring hidden dynamical variables using only rare updates of observables are presented in

SNs 3 and 4, respectively.

Digital twins of parallel RC neural networks for high-dimensional Lorenz-96 climate networks.

We extend the methodology of digital twin to high-dimensional Lorenz-96 climate networks, e.g., $m = 20$. To deal with such a high-dimensional target system, if a single reservoir system is used, the required size of the neural network in the hidden layer will be too large to be computationally efficient. We thus turn to the parallel configuration [25] that consists of many small-size RC networks, each “responsible” for a small part of the target system. For the Lorenz-96 network with $m = 20$ coupled nodes, our digital twin consists of ten parallel RC networks, each monitoring and forecasting the dynamical evolution of two nodes ($D_{\text{out}} = 2$). Because each node in the Lorenz-96 network is coupled to three nearby nodes, we set $D_{\text{in}} = D_{\text{out}} + D_{\text{couple}} = 2 + 3 = 5$ to ensure that sufficient information is supplied to each RC network.

The specific parameters of the digital twin are as follows. The size of the recurrent layer is $D_r = 1,200$. For

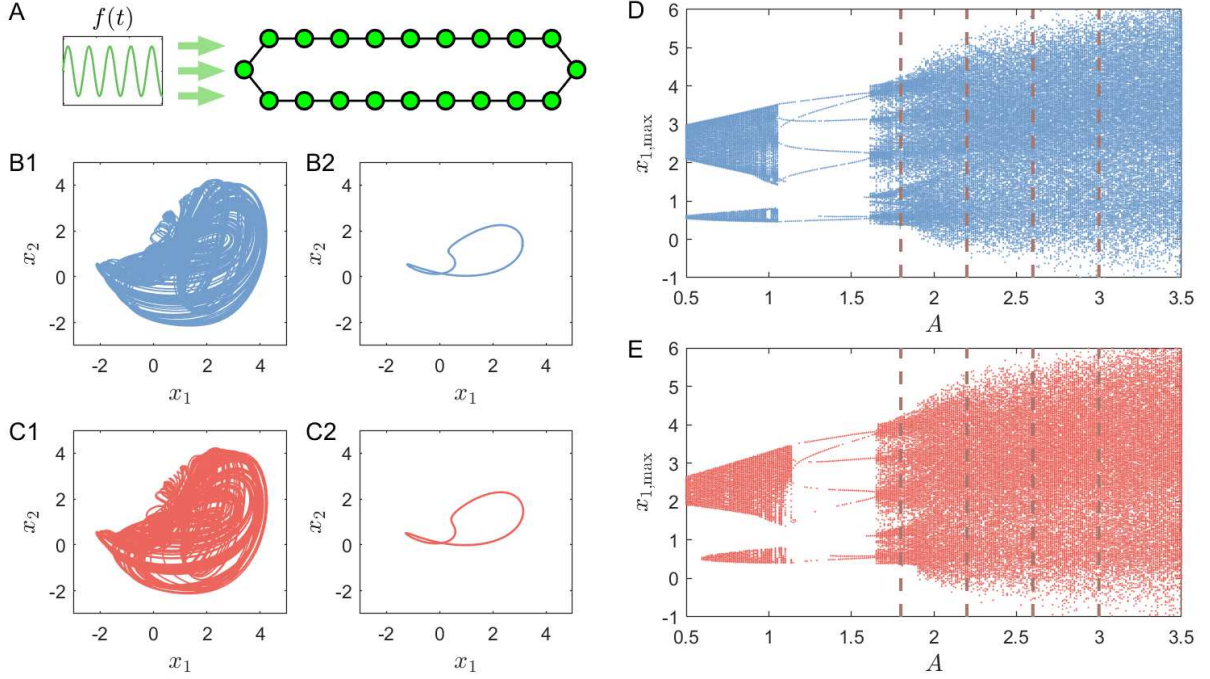


FIG. 2. Digital twin consisting of a number of parallel RC neural networks for high-dimensional chaotic systems. The target system is the Lorenz-96 climate network of $m = 20$ nodes, subject to a global periodic driving $f(t) = A \sin(\omega t) + F$. (A) The structure of the digital twin, where each filled green circle represents a small RC network with the input dimension $D_{in} = 5$ and output dimension $D_{out} = 2$. (B1, B2) A chaotic and periodic attractor in a two-dimensional subspace of the target system for $A = 1.8$ and $A = 1.6$, respectively, for $\omega = 2$ and $F = 2$. (C1, C2) The attractors generated by the digital twin corresponding to those in (B1, B2), respectively, where the training is done using four time series from four different values of forcing amplitude A , all in the chaotic regime. The digital twin with a parallel structure is able to successfully extrapolate the unseen periodic behavior with completely chaotic training data. (D, E) The true and digital-twin generated bifurcation diagrams, respectively, where the four vertical dashed lines in (c) specify the four values of A from which the training time series are obtained. The remarkable agreement between the two bifurcation diagrams indicates that the digital twin so trained can faithfully generate the dynamical behaviors of the high-dimensional target system.

each training value of the forcing amplitude A , the training and validation lengths are $t = 3,500$ and $t = 100$, respectively. The “warming up” length is $t = 20$ and the time step of the dynamical evolution of the digital twin is $\Delta t = 0.025$. The optimized hyperparameter values are $d = 31$, $\lambda = 0.75$, $k_{in} = 0.16$, $k_c = 0.16$, $\alpha = 0.33$, $\beta = 1 \times 10^{-12}$, and $\sigma_{noise} = 10^{-2}$.

The periodic signal used to drive the Lorenz-96 climate network of 20 nodes is $f(t) = A \sin(\omega t) + F$ with $\omega = 2$, and $F = 2$. The structure of the digital twin consists of 20 small RC networks as illustrated in Fig. 2(A). Figures 2(B1) and 2(B2) show a chaotic and a periodic attractor for $A = 1.8$ and $A = 1.6$, respectively, in the (x_1, x_2) plane. Training of the digital twin is conducted by using four time series from four different values of A , all in the chaotic regime. The attractors generated by the digital twin for $A = 1.8$ and $A = 1.6$ are shown in Figs. 2(C1) and 2(C2), respectively, which agree well with the ground truth. Figure 2(D) shows the bifurcation diagram of the target system (the ground truth), where the four values of A : $A = 1.8, 2.2, 2.6$, and 3.0 , from which the training chaotic time series are obtained,

are indicated by the four respective vertical dashed lines. The bifurcation diagram generated by the digital twin is shown in Fig. 2(E), which agrees well with the ground truth in Fig. 2(D).

Digital twins under external driving with varying waveform. The external driving signal is an essential ingredient in our articulation of the digital twin, which is particularly relevant to critical systems of interest such as the climate systems. In applications, the mathematical form of the driving signal may change with time. Can a digital twin produce the correct system behavior under a driving signal that is different than the one it has “seen” during the training phase? Note that, in the examples treated so far, it has been demonstrated that our digital twin can extrapolate the dynamical behavior of a target system under a driving signal of the same mathematical form but with a different amplitude. Here, the task is more challenging as the form of the driving signal has changed.

As a concrete example, we consider the Lorenz-96 climate network of $m = 6$ nodes, where a digital twin is trained with a purely sinusoidal signal $f(t) = A \sin(\omega t) +$

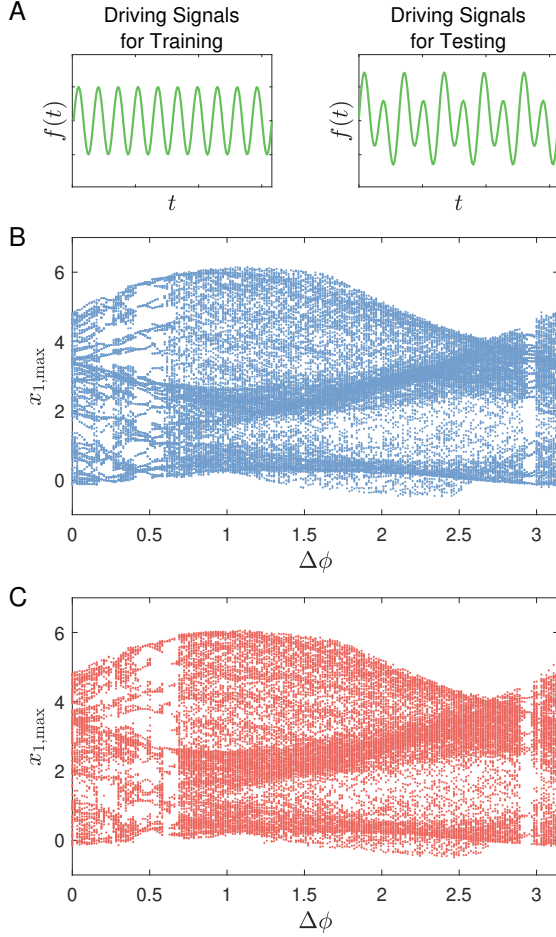


FIG. 3. Effects of waveform change in the external driving on the performance of the digital twin. The time series used to train the digital twin are from the target system subject to external driving of a particular waveform. A change in the waveform occurs subsequently, leading to a different driving signal during the testing phase. (A) During the training phase, the driving signal is of the form $f(t) = A \sin(\omega t) + F$ and time series from four different values of A are used for training the digital twin. The right panel illustrates an example of the changed driving signal during the testing phase. (B) The true bifurcation diagram of the target system under a testing driving signal. (C) The bifurcation diagram generated by the digital twin, facilitated by an optimal level of training noise determined through hyperparameter optimization.

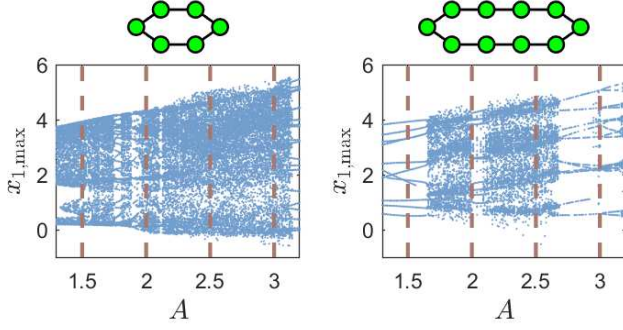
F , as illustrated in the left column of Fig. 3(A). During the testing phase, the driving signal has the form of the sum of two sinusoidal signals with different frequencies: $f(t) = A_1 \sin(\omega_1 t) + A_2 \sin(\omega_2 t + \Delta\phi) + F$, as illustrated in the right panel of Fig. 3(A). We set $A_1 = 2$, $A_2 = 1$, $\omega_1 = 2$, $\omega_2 = 1$, $F = 2$, and use $\Delta\phi$ as the bifurcation parameter. The RC parameter setting is the same as that in Fig. 1. The training and validating lengths for each driving amplitude A value are $t = 3,000$ and $t = 12$, respectively. We find that this setting prevents the digital twin from generating an accurate bifurcation

diagram, but a small amount of dynamical noise to the target system can improve the performance of the digital twin. To demonstrate this, we apply an additive noise term to the driving signal $f(t)$ in the training phase: $df(t)/dt = \omega A \cos(\omega t) + \delta_{\text{DN}} \xi(t)$, where $\xi(t)$ is a Gaussian white noise of zero mean and unit variance, and δ_{DN} is the noise amplitude (e.g., $\delta_{\text{DN}} = 3 \times 10^{-3}$). We use the 2nd-order Heun method [44] to solve the stochastic differential equations describing the target Lorenz-96 system. Intuitively, the noise serves to excite different modes of the target system to instill richer information into the training time series, making the process of learning the target dynamics more effective. Figures 3(B) and 3(C) show the actual and digital-twin generated bifurcation diagrams. Although the digital twin encountered driving signals in a completely “uncharted territory,” it is still able to generate the bifurcation diagram with a reasonable accuracy. Additional results with varying testing waves $f(t)$ are presented in SN 5.

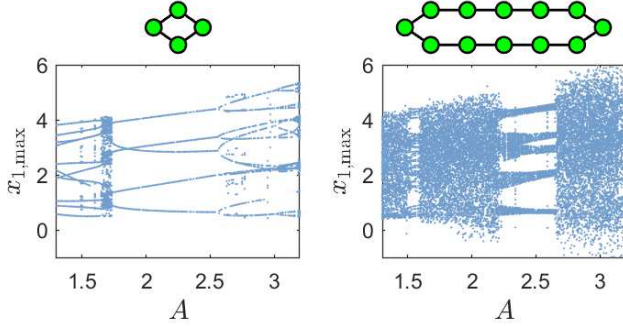
Extrapolability of digital twin with respect to system size. In the examples studied so far, it has been demonstrated that our RC based digital twin has a strong extrapolability in certain dimensions of the parameter space. Specifically, the digital twin trained with time series data from one parameter region can follow the dynamical evolution of the target system in a different parameter regime. One question is whether the digital twin possesses certain extrapolability in the system size. For example, consider the Lorenz-96 climate network of size m . In Fig. 2, we use an array of parallel RC networks to construct a digital twin for the climate network of a fixed size m , where the number of parallel RCs is $m/2$ (assuming that m is even), and training and testing/monitoring are carried out for the same system size. We ask, if a digital twin is trained for climate networks of certain sizes, will it have the ability to generate the correct dynamical behaviors for climate networks of different sizes? If yes, we say that the digital twin has the extrapolability with respect to system size.

As an example, we create a digital twin with time series data from Lorenz-96 climate networks of sizes $m = 6$ and $m = 10$, as shown in Fig. 4(A). For each system size, four values of the forcing amplitude A are used to generate the training time series: $A = 1.5, 2.0, 2.5$, and 3.0 , as marked by the vertical orange dashed lines in Figs. 4(A) and 4(B). As in Fig. 2, the digital twin consists of $m/2$ parallel RC networks, each of size $D_r = 1,500$. The optimized hyperparameter values are determined to be $d = 927$, $\lambda = 0.71$, $k_{\text{in}} = 0.076$, $k_c = 0.078$, $\alpha = 0.27$, $\beta = 1 \times 10^{-11}$, and $\sigma_{\text{noise}} = 3 \times 10^{-3}$. Then we consider climate networks of two different sizes: $m = 4$ and $m = 12$, and test if the trained digital twin can be adapted to the new systems. For the network of size $m = 4$, we keep only two parallel RC networks for the digital twin. For $m = 12$, we add one additional RC network to the trained digital twin for $m = 10$, so the new twin consists of six parallel RC networks of the same hyperparameter values. The true bifurcation diagrams for the climate system of

A Training Data



B Testing Targets



C Testing Results

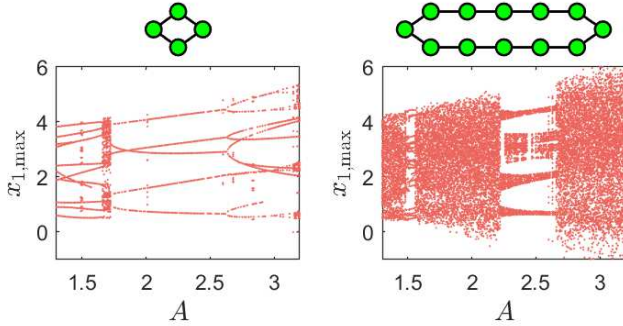


FIG. 4. Demonstration of extrapolability of digital twin in system size. (A) The digital twin is trained using time series from the Lorenz-96 climate networks of size $m = 6$ and $m = 10$. The target climate system is subject to a sinusoidal driving $f(t) = A \sin(\omega t) + F$, and the training time series data are from the A values marked by the eight vertical orange dashed lines. (B) The true bifurcation diagrams of the target climate network of size $m = 4$ and $m = 12$. (C) The corresponding digital-twin generated bifurcation diagrams, where the twin consists of $m/2$ parallel RC networks, each taking input from two nodes in the target system and from the nodes in the network that are coupled to the two nodes.

sizes $m = 4$ and $m = 12$ are shown in Fig. 4(B) (the left and right panels, respectively). The corresponding bifurcation diagrams generated by the adapted digital twins are shown in Fig. 4(C), which agree with the ground truth reasonably well, demonstrating that our RC based digital twin possesses certain extrapolability in system size.

DISCUSSION

We have articulated the principle of creating digital twins for nonlinear dynamical systems based on RCs that are recurrent neural networks. In general, RC is a powerful neural network framework that does not require back-propagation during training but only a linear regression is needed. This feature makes the development of digital twins based on RC computationally efficient. We have demonstrated that a well-trained RC network is able to serve as a digital twin for systems subject to external, time-varying driving. The twin can be used to anticipate possible critical transitions or regime shifts in the target system as the driving force changes, thereby providing early warnings for potential catastrophic collapse of the system. We have used a variety of examples from different fields to demonstrate the workings and the anticipating power of the digital twin, which include the Lorenz-96 climate network of different sizes (in the main text), a driven chaotic CO_2 laser system (SN 1), and an ecological system (SN 2). For low-dimensional nonlinear dynamical systems, a single RC network is sufficient for the digital twin. For high-dimensional systems such as the climate network of a relatively large size, parallel RC networks can be integrated to construct the digital twin. At the level of the detailed state evolution, our recurrent neural network based digital twin is essentially a dynamical twin system that evolves in parallel to the real system, and the evolution of the digital twin can be corrected from time to time using sparse feedback of data from the target system (SN 3). In cases where direct measurements of the target system are not feasible or are too costly, the digital twin provides a way to assess the dynamical evolution of the target system. At the qualitative level, the digital twin can faithfully reproduce the attractors of the target system, e.g., chaotic, periodic, or quasiperiodic, without the need of state updating. In addition, we show that the digital twin is able to accurately predict a critical bifurcation point and the average lifetime of transient chaos that occurs after the bifurcation, even under a driving signal that is different from that during the training (SN 6). The issue of robustness against dynamical and observational noises in the training data has also been treated (SN 7).

To summarize, our RC based digital twins are capable of performing the following tasks: (1) extrapolating certain dynamical evolution of the target system beyond the training parameter regime, (2) making long-term continual forecasting of nonlinear dynamical systems under nonstationary external driving with sparse state updates, (3) inferring the existence of hidden variables in the system and reproducing/predicting their dynamical evolution, (4) adapting to external driving of different waveforms, and (5) extrapolating the global bifurcation behaviors to systems of different sizes.

Our design of the digital twins for nonlinear dynamical systems can be extended in a number of ways.

1. *Online learning.* Online or continual learning is a recent trend in machine-learning research. Unlike the approach of batch learning, where one gathers all the training data in one place and does the training on the entire data set (the way by which training is conducted for our work), in an online learning environment, one evolves the machine learning model incrementally with the flow of data. For each training step, only the newest inputted training data is used to update the machine learning model. When a new data set is available, it is not necessary to train the model over again on the entire data set accumulated so far, but only on the new set. This can result in a significant reduction in the computational complexity. Previously, an online learning approach to RC known as the FORCE learning was developed [45]. An attempt to deal with the key problem of online learning termed “catastrophic forgetting” was made in the context of RC [46]. Further investigation is required to see if these methods can be exploited for creating digital twins through online learning.

2. *Beyond reservoir computing.* Second, the potential power of recurrent neural network based digital twin may be further enhanced by using more sophisticated recurrent neural network models depending on the target problem. We use the RC networks because they are relatively simple yet powerful enough for both low- and high-dimensional dynamical systems. Schemes such as knowledge-based hybrid RC [47] or ODE-nets [48] are worth investigating.

3. *Reinforcement learning.* Is it possible to use digital twins to make reinforcement learning feasible in situations where the target system cannot be “disturbed”? Particularly, reinforcement learning requires constant interaction with the target system during training so that the machine can learn from its mistakes and successes. However, for a real-world system, these interactions may be harmful, uncontrollable, and irreversible. As a result, reinforcement learning algorithms are rarely applied to safety-critical systems [49]. In this case, digital twins can be beneficial. By building a digital twin, the reinforcement learning model does not need to interact with the real system, but with its simulated replica for efficient training. This area of research is called model-based reinforcement learning [50].

4. *Potential benefits of noise.* A phenomenon uncovered in our study is the beneficial role of dynamical noise in the target system. As briefly discussed in Fig. 3, adding dynamic noise in the training dataset enhances the digital twin’s ability to extrapolate the dynamics of the target system with different waveforms of driving. Intuitively, noise can facilitate the exploration of the phase space of the target nonlinear system. A systematic study of the interplay between dynamical noise and the performance of the digital twin is worthy.

5. *Extrapolability.* The demonstrated extrapolability of our digital twin, albeit limited, may open the door to forecasting the behavior of large systems using twins trained on small systems. Much research is needed to

address this issue.

6. *Spatiotemporal dynamical systems with multistability.* We have considered digital twins for a class of coupled dynamical systems: the Lorenz-96 climate model. When developing digital twins for spatiotemporal dynamical systems, two issues can arise. One is the computational complexity associated with such high-dimensional systems. We have demonstrated that parallel reservoir computing provides a viable solution. Another issue is multistability. Spatiotemporal dynamical systems in general exhibit extremely rich dynamical behaviors such as chimera states [51–59]. To develop digital twins of spatiotemporal dynamical systems with multiple coexisting states requires that the underlying recurrent neural networks possess certain memory capabilities. To develop methods to incorporate memories into digital twins is a problem of current interest.

METHODS

The basic construction of the digital twin of a nonlinear dynamical system [61] is illustrated in Fig. 5. It is essentially a recurrent RC neural network with a control mechanism, which requires two types of input signals: the observational time series for training and the control signal $f(t)$ that remains in both the training and self-evolving phase. The hidden layer hosts a random or complex network of artificial neurons. During the training, the hidden recurrent layer is driven by both the input signal $\mathbf{u}(t)$ and the control signal $f(t)$. The neurons in the hidden layer generate a high-dimensional nonlinear response signal. Linearly combining all the responses of these hidden neurons with a set of trainable and optimizable parameters yields the output signal. Specifically, the digital twin consists of four components: (i) an input subsystem that maps the low-dimensional (D_{in}) input signal into a (high) D_r -dimensional signal through the weighted $D_r \times D_{\text{in}}$ matrix $\mathcal{W}_{\text{in}}$, (ii) a reservoir network of N neurons characterized by $\mathcal{W}_r$, a weighted network matrix of dimension $D_r \times D_r$, where $D_r \gg D_{\text{in}}$, (iii) an readout subsystem that converts the D_r -dimensional signal from the reservoir network into an D_{out} -dimensional signal through the output weighted matrix $\mathcal{W}_{\text{out}}$, and (iv) a controller with the matrix $\mathcal{W}_c$. The matrix $\mathcal{W}_r$ defines the structure of the reservoir neural network in the hidden layer, where the dynamics of each node are described by an internal state and a nonlinear hyperbolic tangent activation function.

The matrices $\mathcal{W}_{\text{in}}$, $\mathcal{W}_c$, and $\mathcal{W}_r$ are generated randomly prior to training, whereas all elements of $\mathcal{W}_{\text{out}}$ are to be determined through training. Specifically, the state updating equations for the training and self-

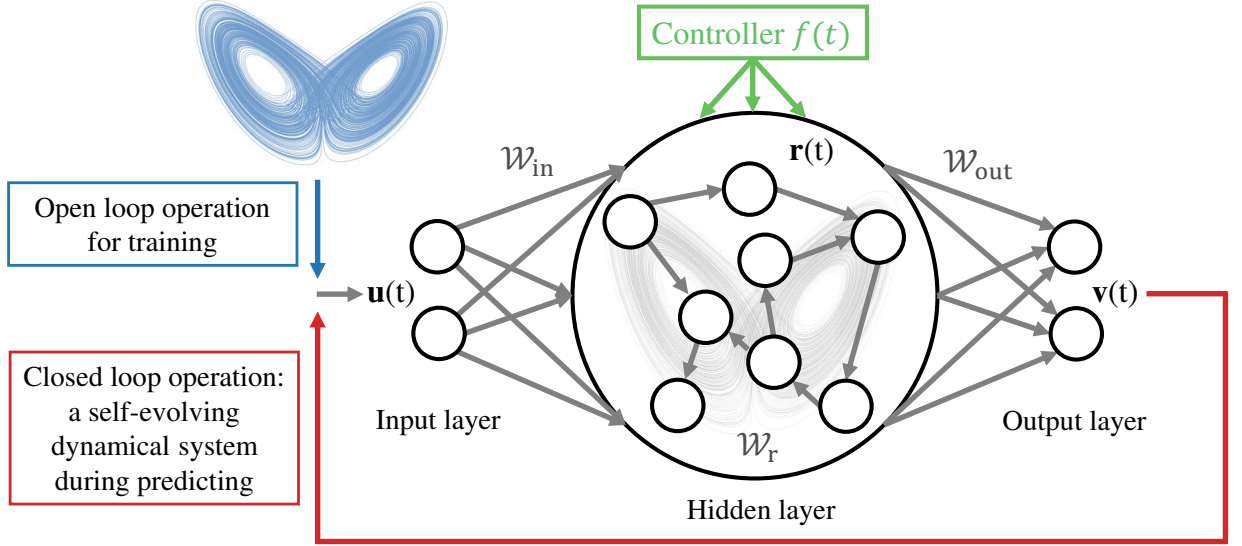


FIG. 5. Basic structure of the digital twin of a chaotic system. It consists of three layers: the input layer, the hidden recurrent layer, an output layer, as well as a controller component. The input matrix $\mathcal{W}_{in}$ maps the D_{in} -dimensional input chaotic data to a vector of much higher dimension D_r , where $D_r \gg D_{in}$. The recurrent hidden layer is characterized by the $D_r \times D_r$ weighted matrix $\mathcal{W}_r$. The dynamical state of the i^{th} neuron in the reservoir is r_i , for $i = 1, \dots, D_r$. The hidden-layer state vector is $\mathbf{r}(t)$, which is an embedding of the input [60]. The output matrix $\mathcal{W}_{out}$ readout the hidden state into the D_{out} -dimensional output vector. The controller provides an external driving signal $f(t)$ to the neural network. During training, the vector $\mathbf{u}(t)$ is the input data, and the blue arrow exists during the training phase only. In the predicting phase, the output vector $\mathbf{v}(t)$ is directly fed back to the input layer, generating a closed-loop, self-evolving dynamical system, as indicated by the red arrow connecting $\mathbf{v}(t)$ to $\mathbf{u}(t)$. The controller remains on in both the training and predicting phases.

evolving phases are, respectively,

$$\begin{aligned} \mathbf{r}(t+\Delta t) &= (1 - \alpha)\mathbf{r}(t) \\ &+ \alpha \tanh[\mathcal{W}_r \mathbf{r}(t) + \mathcal{W}_{in} \mathbf{u}(t) + \mathcal{W}_c f(t)], \end{aligned} \quad (2)$$

$$\begin{aligned} \mathbf{r}(t+\Delta t) &= (1 - \alpha)\mathbf{r}(t) \\ &+ \alpha \tanh[\mathcal{W}_r \mathbf{r}(t) + \mathcal{W}_{in} \mathcal{W}_{out} \mathbf{r}'(t) + \mathcal{W}_c f(t)], \end{aligned} \quad (3)$$

where $\mathbf{r}(t)$ is the hidden state, $\mathbf{u}(t)$ is the vector of input training data, Δt is the time step, the vector $\tanh(\mathbf{p})$ is defined to be $[\tanh(p_1), \tanh(p_2), \dots]^T$ for a vector $\mathbf{p} = [p_1, p_2, \dots]^T$, and α is the leakage factor. During the training, several trials of data are typically used under different driving signals so that the digital twin can “sense, learn, and mingle” the responses of the target system to gain the ability to extrapolate a response to a new driving signal that has never been encountered before. We input these trials of training data, i.e., a few pairs of $\mathbf{u}(t)$ and the associated $f(t)$, through the matrices $\mathcal{W}_{in}$ and $\mathcal{W}_c$ sequentially. Then we record the state vector $\mathbf{r}(t)$ of the neural network during the entire training phase as a matrix $\mathcal{R}$. We also record all the desired output, which is the one-step prediction result $\mathbf{v}(t) = \mathbf{u}(t + \Delta t)$, as the matrix $\mathcal{V}$. To make the readout nonlinear and to avoid unnecessary symmetries in the system [24, 62], we change the matrix $\mathcal{R}$ into $\mathcal{R}'$ by squaring the entries of even dimensions in the states of the hidden layer. [The vector $(\mathbf{r}'(t))$ in Eq. (3) is defined in a similar way.] We carry out a linear regression between $\mathcal{V}$ and $\mathcal{R}'$, with a

ℓ_2 regularization coefficient β , to determine the readout matrix:

$$\mathcal{W}_{out} = \mathcal{V} \cdot \mathcal{R}'^T (\mathcal{R}' \cdot \mathcal{R}'^T + \beta \mathcal{I})^{-1}. \quad (4)$$

To achieve acceptable learning performance, optimization of hyperparameters is necessary. The four widely used global optimization methods are genetic algorithm [63–65], particle swarm optimization [66, 67], Bayesian optimization [68, 69], and surrogate optimization [70–72]. We use the surrogate optimization (the algorithm *surrogateopt* in Matlab). The hyperparameters that are optimized include d - the average degree of the recurrent network in the hidden layer, ρ - the spectral radius of the recurrent network, k_{in} - the scaling factor of $\mathcal{W}_{in}$, k_c - the scaling of $\mathcal{W}_c$, c_0 - the bias in Eq. (2) and (3), α - the leakage factor, and β - the ℓ_2 regularization coefficient.

DATA AVAILABILITY

All relevant data are available from the authors upon request.

CODE AVAILABILITY

All relevant computer codes are available from the authors upon request.

ACKNOWLEDGMENT

We thank Z.-M. Zhai for discussions. This work was supported by the Army Research Office through Grant No. W911NF-21-2-0055 and by the U.S.-Israel Energy Center managed by the Israel-U.S. Binational Industrial Research and Development (BIRD) Foundation.

AUTHOR CONTRIBUTIONS

All authors designed the research project, the models, and methods. L.-W.K. performed the computations. All analyzed the data. L.-W.K. and Y.-C.L wrote the paper.

COMPETING INTERESTS

The authors declare no competing interests.

CORRESPONDENCE

To whom correspondence should be addressed. E-mail: Ying-Cheng.Lai@asu.edu.

-
- [1] Eric J. Tuegel, E. J., Ingraffea, A. R., Eason, T. G. & Spottswood, S. M. Reengineering aircraft structural life prediction using a digital twin. *Int. J. Aerospace Eng.* **2011**, 154798 (2011).
 - [2] Tao, F. & Qi, Q. Make more digital twins. *Nature* **573**, 274–277 (2019).
 - [3] Rasheed, A., San, O. & Kvamsdal, T. Digital twin: Values, challenges and enablers from a modeling perspective. *IEEE Access* **8**, 21980–22012 (2020).
 - [4] Bruynseels, K., de Sio, F. S. & van den Hoven, J. Digital twins in health care: Ethical implications of an emerging engineering paradigm. *Front. Gene.* **9**, 31 (2018).
 - [5] Schwartz, S. M., Wildenhaus, K., Bucher, A. & Byrd, B. Digital twins and the emerging science of self: Implications for digital health experience design and “small” data. *Front. Comp. Sci.* **2**, 31 (2020).
 - [6] Laubenbacher, R., Sluka, J. P. & Glazier, J. A. Using digital twins in viral infection. *Science* **371**, 1105–1106 (2021).
 - [7] Voosen, P. Europe builds ‘digital twin’ of earth to hone climate forecasts. *Science* **370**, 16–17 (2020).
 - [8] Bauer, P., Stevens, B. & Hazeleger, W. A digital twin of earth for the green transition. *Nat. Clim. Change* **11**, 80–83 (2021).
 - [9] Lai, Y.-C. & Tél, T. *Transient Chaos - Complex Dynamics on Finite Time Scales* (Springer, New York, 2011).
 - [10] McCann, K. & Yodzis, P. Nonlinear dynamics and population disappearances. *Ame. Naturalist* **144**, 873–879 (1994).
 - [11] Hastings, A. *et al.* Transient phenomena in ecology. *Science* **361**, eaat6412 (2018).
 - [12] Dhamala, M. & Lai, Y.-C. Controlling transient chaos in deterministic flows with applications to electrical power systems and ecology. *Phys. Rev. E* **59**, 1646–1655 (1999).
 - [13] Lai, Y.-C., Grebogi, C. & Kurths, J. Modeling of deterministic chaotic systems. *Phys. Rev. E* **59**, 2907–2910 (1999).
 - [14] Lai, Y.-C. & Grebogi, C. Modeling of coupled chaotic oscillators. *Phys. Rev. Lett.* **82**, 4803–4806 (1999).
 - [15] Wang, W.-X., Yang, R., Lai, Y.-C., Kovanis, V. & Grebogi, C. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Phys. Rev. Lett.* **106**, 154101 (2011).
 - [16] Wang, W.-X., Lai, Y.-C. & Grebogi, C. Data based identification and prediction of nonlinear and complex dynamical systems. *Phys. Rep.* **644**, 1–76 (2016).
 - [17] Lai, Y.-C. Finding nonlinear system equations and complex network structures from data: A sparse optimization approach. *Chaos* **31**, 082101 (2021).
 - [18] Jaeger, H. The “echo state” approach to analysing and training recurrent neural networks-with an erratum note. *German National Research Center for Information Technology GMD Technical Report* **148**, 13 (2001).
 - [19] Mass, W., Nachtschlaeger, T. & Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neur. Comp.* **14**, 2531–2560 (2002).
 - [20] Jaeger, H. & Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **304**, 78–80 (2004).
 - [21] Haynes, N. D., Soriano, M. C., Rosin, D. P., Fischer, I. & Gauthier, D. J. Reservoir computing with a single time-delay autonomous Boolean node. *Phys. Rev. E* **91**, 020801 (2015).
 - [22] Larger, L. *et al.* High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification. *Phys. Rev. X* **7**, 011015 (2017).
 - [23] Pathak, J., Lu, Z., Hunt, B., Girvan, M. & Ott, E. Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos* **27**, 121102 (2017).
 - [24] Lu, Z. *et al.* Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos* **27**, 041102 (2017).
 - [25] Pathak, J., Hunt, B., Girvan, M., Lu, Z. & Ott, E. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.* **120**, 024102 (2018).
 - [26] Carroll, T. L. Using reservoir computers to distinguish chaotic signals. *Phys. Rev. E* **98**, 052209 (2018).
 - [27] Nakai, K. & Saiki, Y. Machine-learning inference of fluid variables from data using reservoir computing. *Phys.*

- Rev. E* **98**, 023111 (2018).
- [28] Roland, Z. S. & Parlitz, U. Observing spatio-temporal dynamics of excitable media using reservoir computing. *Chaos* **28**, 043118 (2018).
 - [29] Griffith, A., Pomerance, A. & Gauthier, D. J. Forecasting chaotic systems with very low connectivity reservoir computers. *Chaos* **29**, 123108 (2019).
 - [30] Jiang, J. & Lai, Y.-C. Model-free prediction of spatiotemporal dynamical systems with recurrent neural networks: Role of network spectral radius. *Phys. Rev. Research* **1**, 033056 (2019).
 - [31] Tanaka, G. *et al.* Recent advances in physical reservoir computing: A review. *Neu. Net.* **115**, 100–123 (2019).
 - [32] Fan, H., Jiang, J., Zhang, C., Wang, X. & Lai, Y.-C. Long-term prediction of chaotic systems with machine learning. *Phys. Rev. Research* **2**, 012080 (2020).
 - [33] Zhang, C., Jiang, J., Qu, S.-X. & Lai, Y.-C. Predicting phase and sensing phase coherence in chaotic systems with machine learning. *Chaos* **30**, 083114 (2020).
 - [34] Klos, C., Kossio, Y. F. K., Goedeke, S., Gilra, A. & Memmesheimer, R.-M. Dynamical learning of dynamics. *Phys. Rev. Lett.* **125**, 088103 (2020).
 - [35] Kong, L.-W., Fan, H.-W., Grebogi, C. & Lai, Y.-C. Machine learning prediction of critical transition and system collapse. *Phys. Rev. Research* **3**, 013090 (2021).
 - [36] Patel, D., Canaday, D., Girvan, M., Pomerance, A. & Ott, E. Using machine learning to predict statistical properties of non-stationary dynamical processes: System climate, regime transitions, and the effect of stochasticity. *Chaos* **31**, 033149 (2021).
 - [37] Kim, J. Z., Lu, Z., Nozari, E., Pappas, G. J. & Bassett, D. S. Teaching recurrent neural networks to infer global temporal structure from local examples. *Nat. Machine Intell.* **3**, 316–323 (2021).
 - [38] Fan, H., Kong, L.-W., Lai, Y.-C. & Wang, X. Anticipating synchronization with machine learning. *Phys. Rev. Research* **3**, 023237 (2021).
 - [39] Kong, L.-W., Fan, H., Grebogi, C. & Lai, Y.-C. Emergence of transient chaos and intermittency in machine learning. *J. Phys. Complexity* **2**, 035014 (2021).
 - [40] Boltt, E. On explaining the surprising success of reservoir computing forecaster of chaos? the universal machine learning dynamical system with contrast to var and dmd. *Chaos* **31**, 013108 (2021).
 - [41] Gauthier, D. J., Boltt, E., Griffith, A. & Barbosa, W. A. Next generation reservoir computing. *Nat. Commun.* **12**, 1–8 (2021).
 - [42] Carroll, T. L. Optimizing memory in reservoir computers. *Chaos* **32**, 023123 (2022).
 - [43] Lorenz, E. N. Predictability: A problem partly solved. In *Proc. Seminar on Predictability*, vol. 1 (1996).
 - [44] Van den Broeck, C., Parrondo, J., Toral, R. & Kawai, R. Nonequilibrium phase transitions induced by multiplicative noise. *Phys. Rev. E* **55**, 4084 (1997).
 - [45] Sussillo, D. & Abbott, L. F. Generating coherent patterns of activity from chaotic neural networks. *Neuron* **63**, 544–557 (2009).
 - [46] Kobayashi, T. & Sugino, T. Continual learning exploiting structure of fractal reservoir computing. In *International Conference on Artificial Neural Networks*, 35–47 (Springer, 2019).
 - [47] Pathak, J. *et al.* Hybrid forecasting of chaotic processes: Using machine learning in conjunction with a knowledge-based model. *Chaos* **28**, 041101 (2018).
 - [48] Chen, R. T., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K. Neural ordinary differential equations. *Adv. Neu. Info. Proc. Sys.* **31** (2018).
 - [49] Berkenkamp, F., Turchetta, M., Schoellig, A. P. & Krause, A. Safe model-based reinforcement learning with stability guarantees. *arXiv preprint arXiv:1705.08551* (2017).
 - [50] Moerland, T. M., Broekens, J. & Jonker, C. M. Model-based reinforcement learning: A survey. *arXiv preprint arXiv:2006.16712* (2020).
 - [51] Kuramoto, Y. & Battogtokh, D. Coexistence of coherence and incoherence in nonlocally coupled phase oscillators. *Nonlin. Phenom. Complex Syst.* **5**, 380–385 (2002).
 - [52] Abrams, D. M. & Strogatz, S. H. Chimera states for coupled oscillators. *Phys. Rev. Lett.* **93**, 174102 (2004).
 - [53] Omelchenko, I., Maistrenko, Y., Hövel, P. & Schöll, E. Loss of coherence in dynamical networks: Spatial chaos and chimera states. *Phys. Rev. Lett.* **106**, 234102 (2011).
 - [54] Tinsley, M. R., Nkomo, S. & Showalter, K. Chimera and phase-cluster states in populations of coupled chemical oscillators. *Nat. Phys.* **8**, 662 (2012).
 - [55] Hagerstrom, A. M. *et al.* Experimental observation of chimeras in coupled-map lattices. *Nat. Phys.* **8**, 658 (2012).
 - [56] Omelchenko, I., Omel’chenko, O. E., Hövel, P. & Schöll, E. When nonlocal coupling between oscillators becomes stronger: Patched synchrony or multichimera states. *Phys. Rev. Lett.* **110**, 224101 (2013).
 - [57] Omelchenko, I., Zakharaova, A., Hövel, P., Siebert, J. & Schöll, E. Nonlinearity of local dynamics promotes multichimeras. *Chaos* **25**, 083104 (2015).
 - [58] Omelchenko, I., Omel’chenko, O. E., Zakharaova, A. & Schöll, E. Optimal design of tweezer control for chimera states. *Phys. Rev. E* **97**, 012216 (2018).
 - [59] Kong, L.-W. & Lai, Y.-C. Scaling law of transient lifetime of chimera states under dimension-augmenting perturbations. *Phys. Rev. Research* **2**, 023196 (2020).
 - [60] Hart, A., Hook, J. & Dawes, J. Embedding and approximation theorems for echo state networks. *Neu. Net.* **128**, 234–247 (2020).
 - [61] The codes of this work are shared at github.com/lw-kong/Digital_Twin_2021.
 - [62] Herteux, J. & R  th, C. Breaking symmetries of the reservoir equations in echo state networks. *Chaos* **30**, 123142 (2020).
 - [63] Goldberg, D. E. *Genetic Algorithms* (Pearson Education India, 2006).
 - [64] Conn, A. R., Gould, N. I. & Toint, P. A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM J. Numer. Anal.* **28**, 545–572 (1991).
 - [65] Conn, A., Gould, N. & Toint, P. A globally convergent lagrangian barrier algorithm for optimization with general inequality constraints and simple bounds. *Math. Comput.* **66**, 261–288 (1997).
 - [66] Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN’95-International Conference on Neural Networks*, vol. 4, 1942–1948 (IEEE, 1995).
 - [67] Mezura-Montes, E. & Coello, C. A. C. Constraint-handling in nature-inspired numerical optimization: past, present and future. *Swarm Evol. Comput.* **1**, 173–194 (2011).
 - [68] Gelbart, M. A., Snoek, J. & Adams, R. P. Bayesian optimization with unknown constraints. *arXiv preprint*

- arXiv:1403.5607* (2014).
- [69] Snoek, J., Larochelle, H. & Adams, R. P. Practical bayesian optimization of machine learning algorithms. In *NeurIPS*, 2951–2959 (2012).
 - [70] Gutmann, H.-M. A radial basis function method for global optimization. *J. Global Optim.* **19**, 201–227 (2001).
 - [71] Regis, R. G. & Shoemaker, C. A. A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. Comput.* **19**, 497–509 (2007).
 - [72] Wang, Y. & Shoemaker, C. A. A general stochastic algorithmic framework for minimizing expensive black box objective functions based on surrogate models and sensitivity analysis. *arXiv preprint arXiv:1410.6271* (2014).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [SI.pdf](#)